IN THE CLAIMS:

Applicants amend claims 1, 6, 7, and 9. All pending claims and their present status are produced below.

1. (Currently Amended) A method for the direct execution of an XML-document in a data processing system, comprising:

defining the local behavior and process for each element of the XML-document;

integrating executable instructions with at least one XML-document or a document type definition (DTD); and

storing intermediate states of the execution process in a memory of the data processing system by dynamically creating and redefining attributes of elements of the XML document, where the intermediate states define intermediate states of the execution of the executable instructions.

2. (Original) The method according to claim 1, further comprising:

(a) integrating executable instructions by defining for each XML element definition and its instances an action made up of executable actions, and actions which are references to either the action defined for one of the components of the element or to an action defined for any other element of the XML document; and

(b) executing an XML-document by executing the action defined for the root of the XML document.

3. (Original) The method according to claim 1, further comprising:

defining a composition of the action for at least one XML-element definition or instance by graphical flow charts.

4. (Original) The method according to claim 1, further comprising:

defining the composition of the action for at least one XML-element definition or instance in textual form.

5. (Original) The method according to claim 1, further comprising:

representing system states in terms of n-dimensional data cubes;

providing an open interface by making the n-dimensional cubes readable and writeable for other programming and database systems; and making data structures and functionalities of other programming and database systems accessible using executable instructions.

6. (Currently Amended) The method according to claim 1, further comprising modules stored in the memory of the data processing system that define a process for each element, where the modules are valid with respect to the following DTD (Document Type Definition), which is also stored in a memory of the data processing system:

```
    <!element module (derived*, expression?, state*, module*>
    <!attlist module              name CDATA #REQUIRED
                              number CDATA "1">
    <!element derived (argument*, expression)>
    <!attlist derived name CDATA>
    <!element argument EMPTY>
    <!attlist argument name CDATA>
    <!element state (action*, transition*)>
    <!attlist state name CDATA>
    <!element transition (expression, path)>
    <!element path (component?)>
    <!attlist path state CDATA "initial">
    <!element component (component?)>
    <!attlist component           name CDATA #REQUIRED
                                  number CDATA "1">
<!element expression (path | self | src | trg |
                                   evalattr | getfirst | getnext |
                                   parent | root | apply | external |
                                   constant>
<!element action (setattr | ifthen | forall | external)>
<!element src EMPTY>
<!element trg EMPTY>
<!element self EMPTY>
<!element evalattr (expression?)>
<!attlist evalattr attribute CDATA #REQUIRED>
```

```
<!element getfirst (expression?)>
<!attlist getfirst attribute CDATA #REQUIRED>
<!element getnext (expression?)>
<!element parent (expression?)>
<!element root EMPTY>
<!element apply (expression, expression?)>
<!attlist apply op CDATA #REQUIRED>
<!element external (expression*)>
<!attlist external name CDATA
language CDATA >
<!element constant EMPTY>
<!attlist constant value CDATA #REQUIRED>
<!element setAttr (expression?, expression)>
<!attlist setAttr attribute CDATA #REQUIRED>
<!element ifthenelse (expression, action*)>
<!element forall (action*)>
<!attlist forall range CDATA "all-elements"
variable CDATA>.
```

7. (Currently amended) A system for use with the method according to one of the preceding claims, comprising:

a server providing services to at least one client by executing at least parts of a XML-document according to a XML-robot specification sent from the client to the server or a server providing services to at least one client by sending a XML-robot specification and a XML-document to the client, such that said service is provided by executing ~~of~~ at least part of the sent document on the client according to the sent XML-robot specification.

8. (Previously presented) An apparatus for use with the method according to claim 1, comprising:

means for receiving from and sending data to a remote computer; means for storing and accessing a XML-document; means for integrating XML-robot specifications with the XML-document and means for executing the integrated document.

9. (Currently amended) An apparatus for use with the ~~method~~ system according to claim 7, further comprising means for graphical display of XML-robot specifications within an advanced visual integrated development environment and means for generating XML-documents representing said XML-robot specifications.

10. (Original) An apparatus according to claim 8 or 9, further comprising means for examining, validating or animating XML-documents or XML-robot specifications.

11. (Currently Amended) An apparatus for the direct execution of XML documents, comprising:

means for graphical display of XML-robot specifications within an advanced visual integrated development environment; and

means for generating animations of the execution process.

12. (Original) A method for the direct execution of XML documents comprising:

providing an execution specification including

a DTD;

graphical flow charts; and

transition rules;

providing an XML document instance including

an XML document;

using the DTD to validate the XML document;

constructing an attributed structure tree;

decorating the attributed structure tree with the graphical flow charts to create a global flow chart; and

executing the global flow chart according to the transition rules to directly execute the XML document.

13. (Original) A computer-readable medium having computer-readable instructions for performing a method for the direct execution of XML, the method comprising:

providing an execution specification including

a DTD;

graphical flow charts; and

transition rules;

providing an XML document instance including an XML document;

using the DTD to validate the XML document;

constructing an attributed structure tree;

decorating the attributed structure tree with the graphical flow charts to create a global flow chart; and

executing the global flow chart according to the transition rules to directly execute the XML document.

14. (Original) A computer-readable medium having computer-readable instructions for performing a method for the direct execution of XML-documents, the method comprising:

defining the local behavior and process for each element of a XML-document;

integrating executable instructions with a document type definition (DTD), an XML-document; and

storing intermediate states by dynamically creating and redefining element attributes.

15. (Original) A system for the execution of an XML document comprising

an interpreter generator having an input and an output, the input operative to receive an XML specification, the interpreter generator operative to produce at the output an interpreter, the interpreter having an input and an output, the input operative to receive an XML document, the interpreter operative to validate the XML document with respect to a general DTD and to execute the XML document .

16. (Original) A system for the execution of an XML document comprising:

a compiler generator having an input and an output, the input operative to receive an XML specification, the compiler generator operative to produce at the output a compiler, the compiler having an input and an output, the input operative to receive a XML document valid with respect to a general DTD, the compiler operative to produce an executable document at the output.

17. (Original) A system for the execution of an XML document comprising:

a first interpreter having an input, the input operative to receive a XML specification:

a second interpreter coupled to the first interpreter, the second interpreter having an input, the input operative to receive a XML document valid with respect to the general DTD, the first interpreter starting a process in the second interpreter, the second interpreter operative to execute the XML document.

18. (Original) A system for the execution of an XML document comprising:

an interpreter having an input, the input operative to receive a XML specification, the interpreter operative to interpret the XML specification;

a compiler coupled to the interpreter, the compiler having an input and an output, the input operative to receive an XML document, the interpreter operative to start the compiler; the compiler operative to generate an executable XML document on the output.

19. (Original) A method for the execution of an XML document comprising

(a) setting a global variable cur to a root reference;

(b) setting the value of a global variable mod to refer to a module element describing the execution behavior of the root;

(c) copying all state and derived elements from the module mod into the element cur, setting the attribute origin of all state and derived elements to cur;

(d) copying the state and derived elements of the sub-modules of module mod into the corresponding components of element cur;

(e) update cur to cur.traverse; and

(f) if cur is undefined then executing the XML document else returning to (a).

20. (Original) The method according to claim 19, wherein executing the XML document comprises:

(i) setting cur to the XML document's root;

(ii) setting a global variable curstate to initial;

(iii) iterating a variable state over all state elements of cur;

(iv) if a name attribute of state matches curstate then setting cur to the value of attribute origin of state else terminate execution;

(v) iterating over all actions inside state;

(vi) resetting cur to its original value; and

(vii) returning to (iii).

21. (Original) A method for the direct execution of an XML-document in a data processing system, comprising:

defining the local behavior and process for each element of the XML-document;

integrating executable instructions with at least one XML-document or a document type definition (DTD); and

storing intermediate states of the execution process in a memory of the data processing system by dynamically creating and redefining elements.